

Attorney Docket No. 5577-320

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re: Peter J. Brittenham et al;

Serial No.: 09/864,608

Filed: May 23, 2001

For: DYNAMIC REDEPLOYMENT OF SERVICES IN A COMPUTING NETWORK

Group Art Unit: 2157

Examiner: Avi M. Gold

Confirmation No.: 3650

January 16, 2008

CERTIFICATION OF TRANSMISSION

I hereby certify that this correspondence is being transmitted via the Office electronic filing system in accordance with § 1.6(a)(4) to the U.S. Patent and Trademark Office on January 16, 2008.

Signature: _____

Tracy Wallace

Mail Stop Appeal Brief-Patent

Commissioner for Patents

Box 1450

Alexandria, VA 22313-1450

SUPPLEMENTAL APPELLANT'S BRIEF ON APPEAL UNDER 37 C.F.R. §41.37

Sir:

This Supplemental Appeal Brief is filed pursuant to the "Notice of Appeal to the Board of Patent Appeals and Interferences" and the "Reasons In Support Of Applicants' Pre-Appeal Brief Request For Review," both filed July 20, 2006, and in response to the "Notification Of Non-Compliant Appeal Brief" of January 7, 2008. Moreover, this Supplemental Appellants' Brief is being filed within one month of the "Notification Of Non-Compliant Appeal Brief". In particular, this paragraph, the "Status of Claims", and the "Summary of Claimed Subject Matter" have been revised responsive to the "Notification Of Non-Compliant Appeal Brief". The remainder of this Supplemental Appellant's Brief is unchanged from the Appellant's Brief that was filed on February 20, 2007.

It is not believed that an extension of time and/or additional fee(s) are required, beyond those that may otherwise be provided for in documents accompanying this paper. In the event, however, that an extension of time is necessary to allow consideration of this paper, such an

extension is hereby petitioned under 37 C.F.R. Sec. 1.136(a). Any additional fees believed to be due may be charged to Deposit Account No. 09-0461.

Real Party In Interest

The real party in interest is assignee IBM Corporation, having a place of business at Armonk, New York.

Related Appeals and Interferences

The Appellant is aware of no appeals or interferences that would be affected by the present appeal.

Status of Claims

Claims 1-29 stand rejected, having been rejected in the final Office Action of April 20, 2006. The Appellant appeals the final rejection of Claims 1-29 from the final Office Action of April 20, 2006 (the "Final Office Action"), which, as of the filing date of this Appeal Brief, remain under consideration. The claims at issue as included in the Appellant's Amendment filed January 26, 2006 (in response to the Office Action of November 2, 2005), are attached hereto as Appendix A. All of Claims 1-29 are currently pending.

Status of Amendments

The attached Appendix A presents the pending claims 1-29 as amended in the Appellant's Amendment of January 26, 2006. All amendments have been entered in the present case, and no amendments have been filed subsequent to the Final Office Action of April 20, 2006.

Summary of Claimed Subject Matter

The Appellant appeals the final rejection of independent Claims 1, 18 and 19 as being unpatentable over U.S. Patent No. 6,363,411 to Dugan et al. ("Dugan") in view of U.S. Patent No. 6,631,512 to Onyeabor ("Onyeabor"). Claim 18 is a means plus function analog of Claim 1, and Claim 19 is a computer program product analog of Claim 1. The Appellant further appeals the final rejection of dependent Claims 2-3 and 20-29 as being separately unpatentable over

Dugan in view of Onyeabor. The remaining dependent claims are patentable at least as depending from a patentable independent claim.

Claim 1 is directed to a method of dynamically redeploying web services in a computing network, Claim 18 is directed to an analogous system for dynamically redeploying web services in a computing network, and Claim 19 is directed to an analogous computer program product for dynamically redeploying web services in a computing network. Redeployment is discussed in the Application as originally filed, for example, at page 30, line 14 to page 33, line 14 with respect to Figure 11.

With respect to recitations of independent Claim 1, Claim 1 recites a method of dynamically redeploying web services in a computing network. More particularly, a redeployment trigger for a selected web service may be received, for example, by issuing an update request from an origin server 290 to a deployment node 260, and the selected web service may include executable code. *See*, Application, page 31, lines 5-6. One or more network locations where the selected web service including executable code has been deployed from its original location at the origin server may be determined, for example, by having the deployment node 260 obtain a list from its registry 270 of the deployment facilitators 230 where the web service was deployed. *See*, Application, page 32, lines 2-3. The selected web service including the executable code may be programmatically removed from the origin server and programmatically replaced at the origin server, for example, by shutting down the web service, updating the service's executable code and meta information, and then starting the service again. *See*, Application, page 31, lines 2-4. The selected web service may be programmatically removed from the network locations and programmatically replaced at the network locations, for example, by the deployment facilitator 230 receiving the updated web service package from the deployment provider 280 and deploying the web service code and meta information in the run-time environment and then starting the service. *See*, Application, page 32, line 18 to page 33, line 2.

With respect to the recitations of dependent Claim 2, a redeployment request may be issued from an origin server 290. *See*, Application, page 31, lines 5-6. With respect to the recitations of dependent Claim 3, which states that the redeployment trigger is sent when the selected web service including the executable code is to be revised, the update process may start

when the deployment provider updates the web service on the origin server 290. *See*, Application, page 31, lines 2-4.

With respect to the recitations of dependent Claims 24, all of the network locations where the selected web service has been deployed may be determined by the deployment node 260 obtaining a list from its registry 270 of the deployment facilitators 230 where the web service was deployed. *See*, Application, page 32, lines 2-3. The selected web service may be programmatically removed from all of the network locations where the web service has been deployed by shutting down the web service, updating the service's executable code and meta information, and then starting the service again. *See*, Application, page 31, lines 2-4. The selected web service may be programmatically replaced at all of the network locations where the web service has been deployed by the deployment facilitator 230 receiving the updated web service package from the deployment provider 280 and deploying the web service code and meta information in the run-time environment and then starting the service. *See*, Application, page 32, line 18 to page 33, line 2. With respect to the recitations of dependent Claim 25, the selected web service may be replaced with an updated web service including updated executable code by the deployment facilitator 230 receiving the updated web service package from the deployment provider 280 and deploying the web service code and meta information in the run-time environment and then starting the service. *See*, Application, page 32, line 18 to page 33, line 2.

With respect to recitations of independent Claim 18, Claim 18 recites a system for dynamically redeploying web services in a computing network. Means for receiving a redeployment trigger for a selected web service including executable code, for example, may be provided by issuing an update request from an origin server 290 to a deployment node 260. *See*, Application, page 31, lines 5-6. Means for determining one or more network locations where the selected web service including executable code has been deployed from its original location at an origin server, for example, may be provided by having the deployment node 260 obtain a list from its registry 270 of the deployment facilitators 230 where the web service was deployed. *See*, Application, page 32, lines 2-3. Means for programmatically removing the selected web service including the executable code from the network locations and the origin server and means for programmatically replacing the selected web service at the network locations and the origin server, for example, may be provided by shutting down the web service, updating the

service's executable code and meta information, and then starting the service again as discussed in the Application at page 31, lines 2-4, and by the deployment facilitator 230 receiving the updated web service package from the deployment provider 280 and deploying the web service code and meta information in the run-time environment and then starting the service as discussed in the Application at page 32, line 18 to page 33, line 2.

Regarding dependent Claim 20 (which depends from Claim 18), a redeployment request may be issued from an origin server 290. *See*, Application, page 31, lines 5-6. Regarding means plus function recitations of dependent Claim 22, means for sending the redeployment trigger when the selected web service is to be revised may be provided by the update process starting when the deployment provider updates the web service on the origin server 290. *See*, Application, page 31, lines 2-4.

Regarding dependent Claim 26 (which depends from Claim 18), the means for determining all of the network locations where the selected web service has been deployed may be provided by the deployment node 260 obtaining a list from its registry 270 of the deployment facilitators 230 where the web service was deployed. *See*, Application, page 32, lines 2-3. The means for programmatically removing the selected web service from all of the network location where the web service has been deployed may be provided by shutting down the web service, updating the service's executable code and meta information, and then starting the service again. *See*, Application, page 31, lines 2-4. Programmatically replacing the selected web service at all of the network locations where the web service has been deployed may be provided by the deployment facilitator 230 receiving the updated web service package from the deployment provider 280 and deploying the web service code and meta information in the run-time environment and then starting the service. *See*, Application, page 32, line 18 to page 33, line 2. Regarding dependent Claim 27 (which depends from Claim 18), the means for replacing the selected web service with an updated web service including updated executable code may be provided by the deployment facilitator 230 receiving the updated web service package from the deployment provider 280 and deploying the web service code and meta information in the run-time environment and then starting the service. *See*, Application, page 32, line 18 to page 33, line 2.

With respect to recitations of independent Claim 19, Claim 19 recites a computer program product for dynamically redeploying web services in a computing network wherein the computer program product is embodied on one or more computer-readable media. In particular, computer-readable program code configured to receive a redeployment trigger for a selected web service including executable code, for example, may issue an update request from an origin server 290 to a deployment node 260. *See*, Application, page 31, lines 5-6. Computer-readable program code configured to determine one or more network locations where the selected web service including executable code has been deployed from its original location at the origin server, for example, may have the deployment node 260 obtain a list from its registry 270 of the deployment facilitators 230 where the web service was deployed. *See*, Application, page 32, lines 2-3. Computer-readable program code configured to programmatically remove the selected web service including the executable code from the origin server and to programmatically replace the selected web service at the origin server, for example, may shut down the web service, update the service's executable code and meta information, and then start the service again. *See*, Application, page 31, lines 2-4. Computer-readable program code configured to programmatically remove the selected web service from the network locations and to programmatically replace the selected web service at the network locations, for example, may have the deployment facilitator 230 receive the updated web service package from the deployment provider 280 and deploy the web service code and meta information in the run-time environment, and may then start the service. *See*, Application, page 32, line 18 to page 33, line 2.

With respect to the recitations of dependent Claim 21, a redeployment request may be issued from an origin server 290. *See*, Application, page 31, lines 5-6. With respect to the recitations of dependent Claim 23, computer-readable program code may also be configured to send the redeployment trigger when the selected web service including the executable code is to be revised by starting the update process when the deployment provider updates the web service on the origin server 290. *See*, Application, page 31, lines 2-4.

With respect to the recitations of dependent Claim 28, all of the network locations where the selected web service has been deployed may be determined by the deployment node 260 obtaining a list from its registry 270 of the deployment facilitators 230 where the web service

was deployed. *See*, Application, page 32, lines 2-3. The selected web service may be programmatically removed from all of the network locations where the web service has been deployed by shutting down the web service, updating the service's executable code and meta information, and then starting the service again. *See*, Application, page 31, lines 2-4. The selected web service may be programmatically replaced at all of the network locations where the web service has been deployed by the deployment facilitator 230 receiving the updated web service package from the deployment provider 280 and deploying the web service code and meta information in the run-time environment and then starting the service. *See*, Application, page 32, line 18 to page 33, line 2. With respect to the recitations of dependent Claim 29, the selected web service may be replaced with an updated web service including updated executable code by the deployment facilitator 230 receiving the updated web service package from the deployment provider 280 and deploying the web service code and meta information in the run-time environment and then starting the service. *See*, Application, page 32, line 18 to page 33, line 2.

Grounds of Rejection To Be Reviewed on Appeal

Independent Claims 1, 18 and 19 and dependent Claims 2-3 and 20-29 stand rejected under 35 U.S.C. §103(a) as being unpatentable over U.S. Patent No. 6,363,411 to Dugan et al. ("Dugan") in view of U.S. Patent No. 6,631,512 to Onyeabor ("Onyeabor").

Independent Claims 1, 18, and 19 are patentable over the combination of Dugan and Onyeabor for at least the reasons discussed below. Dependent Claims 2-17 and 20-29 are patentable at least as per the patentability of independent Claims 1, 18, and 19 from which they depend. In addition, dependent Claims 2-3 and 20-29 are separately patentable over Dugan and Onyeabor for at least the reasons discussed below.

Arguments

I. Introduction to 35 U.S.C. §103 Analysis

Claims 1-29 are rejected under 35 U.S.C. §103(a) as being unpatentable. A determination under §103 that an invention would have been obvious to someone of ordinary skill in the art is a conclusion of law based on fact. *Panduit Corp. v. Dennison Mfg. Co.* 810 F.2d 1593, 1 U.S.P.Q.2d 1593 (Fed. Cir. 1987), *cert. denied*, 107 S.Ct. 2187. After the involved facts are determined, the decision maker must then make the legal determination of whether the claimed invention as a whole would have been obvious to a person having ordinary skill in the art at the time the invention was unknown, and just before it was made. *Id.* at 1596. The United States Patent and Trademark Office (USPTO) has the initial burden under §103 to establish a *prima facie* case of obviousness. *In re Fine*, 837 F.2d 1071, 5 U.S.P.Q.2d 1596, 1598 (Fed. Cir. 1988).

To establish a *prima facie* case of obviousness, the prior art reference or references when combined must teach or suggest *all* the recitations of the claims, and there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. M.P.E.P. §2143. The mere fact that references can be combined or modified does not render the resultant combination obvious unless the prior art also suggests the desirability of the combination. M.P.E.P. §2143.01, citing *In re Mills*, 916 F.2d 680, 16 U.S.P.Q.2d 1430 (Fed. Cir. 1990). As emphasized by the Court of Appeals for the Federal Circuit, to support

combining references, evidence of a suggestion, teaching, or motivation to combine must be **clear and particular**, and this requirement for clear and particular evidence is not met by broad and conclusory statements about the teachings of references. *In re Dembiczak*, 50 U.S.P.Q.2d 1614, 1617 (Fed. Cir. 1999). In another decision, the Court of Appeals for the Federal Circuit has stated that, to support combining or modifying references, there must be **particular** evidence from the prior art as to the reason the skilled artisan, with no knowledge of the claimed invention, would have selected these components for combination in the manner claimed. *In re Kotzab*, 55 U.S.P.Q.2d 1313, 1317 (Fed. Cir. 2000).

Appellants respectfully submit that the pending claims are patentable over the cited references for at least the reason that none of the cited references, either alone or in combination, thereof disclose or suggest all of the recitations of the claims. The patentability of the pending claims is discussed in detail hereinafter.

II. Claims 1-11 and 14-29 Are Patentable Over The Cited Art

Claims 1-11 and 14-29 stand rejected as being unpatentable over Dugan in view of Onyeabor. Dugan and Onyeabor, however, fail to teach or suggest Claims 1-11 and 14-29 for at least the reasons discussed below.

A. Independent Claims 1, 18 and 19 Are Patentable Over The Cited Art

Independent Claims 1, 18 and 19 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Dugan in view of Onyeabor. However, these claims are patentable for at least the reasons discussed below.

Claim 1, for example, recites a method of dynamically redeploying web services in a computing network, the method including:

- receiving a redeployment trigger for a selected web service wherein the selected web service includes executable code;

- determining one or more network locations where the selected web service including the executable code has been deployed from its original location at an origin server;

- programmatically removing the selected web service including the executable code from the network locations and the origin server; and

- programmatically replacing the selected web service at the network locations and the origin server.

The Appellant respectfully submits that the combination of Dugan and Onyeabor fails to teach or suggest removing and replacing a selected web service including executable code in a computing network as recited in Claim 1.

As discussed in the Manual Of Patent Examining Procedure (MPEP), three basic criteria must be met to establish a *prima facie* case of obviousness. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. Second, there must be a reasonable expectation of success. Finally, the prior art reference (or references when combined) must teach or suggest all the claim limitations. Moreover, the teaching or suggestion to make the claimed combination and the reasonable expectation of success must both be found in the prior art, not in applicant's disclosure. *See*, MPEP, Sec. 2143.

In particular, Dugan is related to "telecommunications networks" (col. 1, lines 17-18) as opposed to a computing network as recited in Claim 1. In rejecting Claim 1, the Final Office Action of April 20, 2006 (the Final Office Action), cites Column 20, lines 14-26 of Dugan. Regarding the recitation of "a selected web service", portions of Dugan cited by the Final Office Action state that:

These service node profiles (e.g., Table 1) and service profiles (e.g., Table 2) are input to SA and stored therein to enable automatic tracking of: 1) the capabilities of each service node, i.e., how many computers and SLEE(s), and the resource capacity of each; 2) which services and data are to be deployed to which service nodes and when; and, 3) the configuration of service execution, i.e., at which times an SLP should run persistently versus on-demand, for example. The capabilities of each node and computer in the network is maintained, so that simple and complex business rules governing data/service distribution, data/service activation and data/service removal may be applied to optimize the execution of services on IDNA/NGIN service nodes. (Emphasis added.)

Dugan, col. 20, lines 14-26. The "services" of Dugan, however, are discussed as follows:

The present invention is directed to an intelligent network designed to perform intelligent call processing services for any type of call received at a resource complex or switching platform. (Emphasis added.)

Dugan, col. 5, line 66 to col. 6, line 2. Dugan thus relates to call processing services as opposed to web services.

Accordingly, Dugan fails to teach or suggest redeploying web services in a computing network including programmatically removing a selected web service including executable code and programmatically replacing the selected web service as recited in Claim 1. Moreover, Dugan also fails to teach or suggest receiving a redeployment trigger. In addition, the Final Office Action concedes that:

Dugan fails to teach the limitation further including the use of a selected web service wherein the selected web service includes executable code.

Final Office Action, page 3. For at least the reasons discussed above, Dugan thus fails to teach or suggest:

- 1.) removing a selected web service including executable code;
- 2.) replacing the selected web service; and/or
- 3.) receiving a redeployment trigger.

Onyeabor fails to provide the missing teachings. In support of the rejection, the Final Office Action states that:

Onyeabor teaches Web page development, deployment, and execution conducive to database access and manipulation over the Internet (see Abstract). Onyeabor teaches the use of the deployment of a web page with that web page including executable code (col. 6, lines 14-20, col. 16, lines 35-44).

Office Action, page 3. As discussed in Onyeabor:

the method and apparatus of the present invention include a Web page development tool which enables a developer to create a Web page document which includes executable code, thus eliminating the need to download foreign executables during display and manipulation of a page. This virtually eliminates the risk that malicious code will be downloaded and allowed to wreak havoc on the client machine.

Onyeabor, col. 6, lines 14-20. Onyeabor further states that:

after creation of a Web page by Web page development computer 110, a developer may then request that computer 110 "deploy" the new Web page. "Deployment" of a Web page refers to the act of sending the page to a server which will then provide access to the page to client computers via the Internet. Web page development computer may send the Web page to the server via the Internet, a LAN, a WAN, any other type of optical, wireless, or wired link (or links), or via a tangible data storage medium.

Onyeabor, col. 16, lines 35-44. Onyeabor, however, fails to teach or suggest:

- 1.) removing a selected web service including executable code;
- 2.) replacing the selected web service; and/or

3.) receiving a redeployment trigger.

As neither of the cited references (taken alone or in combination) teaches or suggest any of the three claim recitations noted above, the combination of Dugan and Onyeabor fails to teach or suggest the method of Claim 1. Stated in other words, the combination of Dugan and Onyeabor fails teach or suggest all the claim limitations as required by MPEP Sec. 2143.

In addition, there is no motivation to selectively combine elements of Dugan and Onyeabor to somehow teach or suggest the method of Claim 1 as required by MPEP Sec. 2143. In particular, there is no motivation to combine aspects of Web page development/deployment/execution of Onyeabor (*see*, Onyeabor, col. 1, lines 8-11) with telecommunications service processing of Dugan (*see*, Dugan, col. 1, lines 17-22). Moreover, there is no reasonable expectation that aspects of Web page development/deployment/execution from Onyeabor can be successfully substituted for elements of the telecommunications switching network of Dugan as required by MPEP Sec. 2143.

Accordingly, the Appellant respectfully submits that the combination of Dugan and Onyeabor fails to teach or suggest the recitations of Claim 1 and that Claim 1 is thus patentable. The Appellant further submits that Claims 18 and 19 are patentable for reasons similar to those discussed above with regard to Claim 1. In addition, Dependent Claims 2-17 and 20-23 are patentable at least as per the patentability of Claims 1, 18, and 19 from which they depend.

The Advisory Action of July 6, 2006 (the Advisory Action), states that the recitation "web services in a computing network" in Claim 1 has not been given patentable weight because it occurs in the preamble. *See*, Advisory Action, Continuation Sheet. In response, the Appellant notes that a/the "selected web service" is recited in each step of Claim 1, and that "web services in a computing network" from the preamble of Claim 1 defines a/the "selected web service" as being in the computing network. Accordingly, the indicated recitations in the preamble must be given patentable weight.

The Advisory Action further states that one cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references. *See*, Advisory Action, Continuation Sheet. The Appellant respectfully submits, however, that the prior art references when combined must teach or suggest all the claim limitations (*see* MPEP Sec. 2143), and that neither of the cited references (taken alone or in combination) teaches or

suggests: (1) removing a selected web service including executable code; (2) replacing the selected web service; and/or (3) receiving a redeployment trigger.

In addition, the Advisory Action states that motivation for the combination of Dugan and Onyeabor is found at column 6, lines 19-20 of Onyeabor. *See*, Advisory Action, Continuation Sheet. The cited portion of Onyeabor states that:

This virtually eliminates the risk that malicious code will be downloaded and allowed to wreak havoc on the client machine.

Onyeabor, col. 6, lines 18-20. As set forth above, however, there is no motivation to combine aspects of Web page development/deployment/execution of Onyeabor (*see*, Onyeabor, col. 1, lines 8-11) with telecommunications service processing of Dugan (*see*, Dugan, col. 1, lines 17-22). Moreover, there is no reasonable expectation that aspects of Web page development/deployment/execution from Onyeabor can be successfully substituted for elements of the telecommunications switching network of Dugan.

Accordingly, the Appellant respectfully submits that Dugan and Onyeabor fail to teach or suggest the recitations of Claim 1 and that Claim 1 is thus patentable. The Appellant further submits that Claims 18 and 19 are patentable for reasons similar to those discussed above with regard to Claim 1. In addition, Dependent Claims 2-17, and 20-29 are patentable at least as per the patentability of Claims 1, 18, and 19 from which they depend.

B. Various Dependent Claims Are Independently Patentable

As discussed above, dependent Claims 2-17 and 20-29 are patentable at least as per the patentability of Claims 1, 18, and 19 from which they depend. Various of these dependent claims are also separately patentable for reasons discussed in greater detail below.

i. Claims 2, 20, And 21 Are Separately Patentable Over The Cited Art

Dependent Claims 2, 20, and 21 stand rejected under 35 U.S.C. Sec. 103(a) as being unpatentable over Dugan in view of Onyeabor. Claims 2, 20, and 21 are patentable at least as per the patentability of Claims 1, 18, and 19 from which they depend. Claims 2, 20, and 21 are also separately patentable for at least the additional reasons discussed below.

Dependent Claim 2, for example, depends from Claims 1 and thus includes all recitations of Claim 1 as discussed above. In addition, Claim 2 recites that the redeployment trigger comprises a redeployment request from the origin server. Portions of Dugan cited by the Final Office Action with respect to Claim 2 state that:

These service node profiles (e.g., Table 1) and service profiles (e.g., Table 2) are input to SA and stored therein to enable automatic tracking of: 1) the capabilities of each service node, i.e., how many computers and SLEE(s), and the resource capacity of each; 2) which services and data are to be deployed to which service nodes and when; and, 3) the configuration of service execution, i.e., at which times an SLP should run persistently versus on-demand, for example.

Dugan, col. 20, lines 14-21.

Dugan, however, fails to teach or suggest a redeployment request, much less, a redeployment request from an origin server which is the original location of the web service being replaced (as defined in Claim 1). Moreover, nothing from Onyeabor is cited by the Final Office Action with respect to the additional recitations of Claim 2. The Advisory Action of July 6, 2006 (the Advisory Action) further states that:

Regarding the argument to claims 2 and 3, the applicant argues that the reference, Dugan, does not disclose a redeployment request/trigger. The examiner respectfully disagrees, as seen in, col. 20, lines 14-26, there is the deployment of services and data which require maintenance, thus inherently involving redeployment. In addition on col. 20, line 66 – col. 21, line 7 there are retries of deployed data distribution, which is the same as redeployment.

Advisory Action, Continuation Sheet. Accepting the characterization of Dugan set forth in the Advisory Action for the sake of argument, the Advisory Action and/or the Final Office Action fail to point out any portion of Dugan and/or Onyeabor that teaches or suggests a redeployment trigger/request, much less a redeployment request from an origin server (with the origin server defined in Claim 1 as the location from which the selected web service was deployed).

Moreover, the portions of Dugan cited by the Advisory Action with respect to Claim 2 fail to teach or suggest such a redeployment request from an origin server. While the Advisory Action alleges that Dugan "inherently" involves redeployment, Dugan and/or Onyeabor fail to teach or suggest a redeployment request from an origin server, and the Final Office Action and/or the Advisory Action do not identify any portion of Dugan and/or Onyeabor that provide such a teaching.

For at least the reasons discussed above, Claim 2 is separately patentable over the combination of Dugan and Onyeabor. In addition, Claims 20 and 21 are separately patentable for reasons similar to those discussed above with respect to Claim 2.

ii. Claims 3, 22, And 23 Are Separately Patentable Over The Cited Art

Dependent Claims 3, 22, and 23 stand rejected under 35 U.S.C. Sec. 103(a) as being unpatentable over Dugan in view of Onyeabor. Claims 3, 22, and 23 are patentable at least as per the patentability of Claims 1, 18, and 19 from which they depend. Claims 3, 22, and 23 are also separately patentable for at least the additional reasons discussed below.

Dependent Claim 3, for example, depends from Claims 1 and thus includes all recitations of Claim 1 as discussed above. In addition, Claim 3 recites sending the redeployment trigger when the selected web service including the executable code is to be revised. Portions of Dugan cited by the Final Office Action with respect to Claim 3 state that:

The capabilities of each node and computer in the network is maintained, so that simple and complex business rules governing data/service distribution, data/service activation and data/service removal may be applied to optimize the execution of services on IDNA/NGIN service nodes.

Dugan, col. 20, lines 22-26.

While Dugan discusses business rules governing data/service distribution, activation, and removal, Dugan fails to teach or suggest a redeployment trigger, much less sending a redeployment trigger when a selected web service is to be revised. Moreover, nothing from Onyeabor is cited by the Final Office Action with respect to the additional recitations of Claim 3. The Advisory Action further states that:

Regarding the argument to claims 2 and 3, the applicant argues that the reference, Dugan, does not disclose a redeployment request/trigger. The examiner respectfully disagrees, as seen in, col. 20, lines 14-26, there is the deployment of services and data which require maintenance, thus inherently involving redeployment. In addition on col. 20, line 66 – col. 21, line 7 there are retries of deployed data distribution, which is the same as redeployment.

Advisory Action, Continuation Sheet. Accepting the characterization of Dugan as set forth in the Advisory Action, the Advisory Action and/or the Final Office Action fail to point out any portion of Dugan and/or Onyeabor that teaches or suggests a redeployment trigger, much less a

redeployment trigger that is sent when a selected web service is to be revised. Moreover, the portions of Dugan cited by the Advisory Action with respect to Claim 3 fail to teach or suggest such sending such a redeployment trigger when a selected web service is to be revised. While the Advisory Action alleges that Dugan "inherently" involves redeployment, Dugan and/or Onyeabor fail to teach or suggest sending a redeployment trigger when a selected web service is to be revised, and the Final Office Action and/or the Advisory Action do not identify any portion of Dugan and/or Onyeabor that provide such a teaching.

For at least the reasons discussed above, Claim 3 is separately patentable over the combination of Dugan and Onyeabor. In addition, Claims 22 and 23 are separately patentable for reasons similar to those discussed above with respect to Claim 3.

iii. Claims 24, 26, and 28 Are Separately Patentable Over The Cited Art

Dependent Claims 24, 26, and 28 stand rejected under 35 U.S.C. Sec. 103(a) as being unpatentable over Dugan in view of Onyeabor. Claims 24, 26, and 28 are patentable at least as per the patentability of Claims 1, 18, and 19 from which they depend. Claims 24, 26, and 28 are also separately patentable for at least the additional reasons discussed below.

Dependent Claim 24, for example, depends from Claim 1 and thus includes all recitations of Claim 1 as discussed above. In addition, Claim 24 recites:

wherein determining one or more network locations where the selected web service has been deployed includes determining all of the network locations where the selected web service has been deployed;

wherein programmatically removing the selected web service from the network locations includes programmatically removing the selected web service from all of the network location where the web service has been deployed; and

wherein programmatically replacing the selected web service at the network locations includes programmatically replacing the selected web service at all of the network locations where the web service has been deployed.

In contrast, Dugan discusses a "service support provisioning function ... with rules based on ... load balancing among service nodes, network call routing efficiencies, and service demand." Dugan, col. 20, lines 27-32. Accordingly, Dugan fails to teach or suggest removing a selected web service from all network location where the web service has been deployed and/or replacing the selected web service at all of the network locations where the web service has been deployed.

Moreover, nothing from Onyeabor is cited by the Final Office Action with respect to the additional recitations of Claim 24. The Advisory Action further states that:

Regarding the argument to claim 24, the applicant argues that the reference, Dugan, does not disclose removing a selected web service from all network locations where the web service has been deployed. The Examiner respectfully disagrees, as seen in, col. 20, lines 20-26, there is use of service removal.

Advisory Action, Continuation Sheet. Portions of Dugan cited by the Advisory Action state that:

... the configuration of service execution, i.e., at which times an SLP should run persistently versus on-demand, for example. The capabilities of each node and computer in the network is maintained, so that simple and complex business rules governing data/service distribution, data/service activation and data/service removal may be applied to optimize the execution of services on IDNA/NGIN service nodes. (Underline added.)

Dugan, col. 20, lines 20-26. The Appellant respectfully submits, however, that data/service removal in the telecommunications network of Dugan fails to teach or suggest removing a web service from all network locations where the web service has been deployed, especially where the web service includes executable code (as defined in Claim 1 from which Claim 24 depends).

For at least the reasons discussed above, Claim 24 is separately patentable over the combination of Dugan and Onyeabor. In addition, Claims 26 and 28 are separately patentable for reasons similar to those discussed above with respect to Claim 24.

iv. Claims 25, 27, and 29 Are Separately Patentable Over The Cited Art

Dependent Claims 25, 27, and 29 stand rejected under 35 U.S.C. Sec. 103(a) as being unpatentable over Dugan in view of Onyeabor. Claims 25, 27, and 29 are patentable at least as per the patentability of Claims 1, 18, and 19 from which they depend. Claims 25, 27, and 29 are also separately patentable for at least the additional reasons discussed below.

Dependent Claim 25, for example, depends from Claims 1 and thus includes all recitations of Claim 1 as discussed above. In addition, Claim 25 recites that programmatically replacing the selected web service at the network locations comprises replacing the selected web service with an updated web service including updated executable code. As discussed above with respect to Claim 24, Dugan discusses provisioning based on load balancing, efficiencies, and demand. *See*, Dugan, col. 20, lines 27-32. Portions of Dugan cited by the Final Office Action with respect to Claim 25 discuss tracking of:

2) which services and data are to be deployed to which service nodes and when; and 3) the configuration of service execution, i.e., at which times an SLP should run persistently versus on-demand, for example. The capabilities of each node and computer in the network is maintained, so that simple and complex business rules governing data/service distribution, data/service activation and data/service removal may be applied to optimize the execution of services on IDNA/NGIN service nodes.

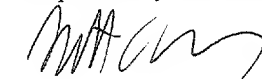
Dugan, col. 20, lines 18-26. Dugan, however, fails to teach or suggest replacing a selected web service with an updated web service including updated executable code. Moreover, nothing from Onyeabor is cited by the Final Action or by the Advisory Action with respect to the additional recitations of Claim 25.

For at least the reasons discussed above, Claim 25 is separately patentable over the combination of Dugan and Onyeabor. In addition, Claims 27 and 29 are separately patentable for reasons similar to those discussed above with respect to Claim 25.

IV. Conclusion

In summary, the Appellant respectfully submits that the cited art fails to teach or suggest all recitations of independent Claims 1, 18, and 19 for at least the reasons discussed above. The remaining dependent claims are patentable at least as depending from patentable independent Claims 1, 18 and 19. Moreover, dependent Claims 2-3 and 20-29 are separately patentable over the cited art for at least the reasons discussed above. Accordingly, the Appellant respectfully requests reversal of the rejection of Claims 1-29 based on the cited references.

Respectfully submitted,



Scott C. Hatfield
Registration No. 38,176
Attorney for the Appellant

USPTO Customer No. 46589
Myers Bigel Sibley & Sajovec, P.A.
P.O. Box 37428
Raleigh, NC 27627
919-854-1400
919-854-1401 (Fax)

Appendix A: Claims

1. (rejected) A method of dynamically redeploying web services in a computing network, the method comprising:
 - receiving a redeployment trigger for a selected web service wherein the selected web service includes executable code;
 - determining one or more network locations where the selected web service including the executable code has been deployed from its original location at an origin server;
 - programmatically removing the selected web service including the executable code from the network locations and the origin server; and
 - programmatically replacing the selected web service at the network locations and the origin server.
2. (rejected) The method according to claim 1, wherein the redeployment trigger comprises a redeployment request from the origin server.
3. (rejected) The method according to claim 1, further comprising:
 - sending the redeployment trigger when the selected web service including the executable code is to be revised.
4. (rejected) The method according to claim 1, further comprising:
 - receiving client requests for the selected web service;
 - serving the received requests from the network locations prior to receiving the redeployment trigger; and
 - serving the received requests using the replaced web service after programmatically removing the selected web service and programmatically replacing the selected web service.
5. (rejected) The method according to claim 1, further comprising:
 - unpublishing the selected web service after receiving the redeployment trigger, until completion of programmatically removing the selected web service and programmatically replacing the selected web service, and then republishing the selected web service thereafter.

6. (rejected) The method according to claim 2, further comprising:
sending a subsequent redeployment request to each of the network locations, responsive to receiving the redeployment request from the origin server.

7. (rejected) The method according to claim 6, wherein programmatically removing the selected web service further comprises:
receiving the subsequent redeployment request at a selected one of the network locations;
programmatically shutting down the selected web service at the selected one, responsive to receiving the subsequent redeployment request; and
programmatically removing the executable code which implements the selected web service from a run-time environment of the selected one, subsequent to the programmatically shutting down.

8. (rejected) The method according to claim 6, wherein programmatically replacing the selected web service further comprises:
issuing a deployment request for the selected web service from a selected one of the network locations;
receiving a response message at the selected one of the network locations, the response message containing a replacement for the selected web service; and
deploying the replacement for the selected web service at the selected one of the network locations.

9. (rejected) The method according to claim 8, wherein the deployment request comprises a web service description of the selected web service encoded in a standardized web service description notation.

10. (rejected) The method according to claim 9, wherein the web service description comprises an interface definition of a dynamic deployment web service and an implementation definition of the dynamic deployment web service.

11. (rejected) The method according to claim 10, wherein the dynamic deployment web service resides on the origin server.

12. (rejected) The method according to claim 11, wherein the issued deployment request comprises a SOAP ("Simple Object Access Protocol") request.

13. (rejected) The method according to claim 11, wherein the issued deployment request comprises an XML ("Extensible Markup Language") Protocol request.

14. (rejected) The method according to claim 11, wherein the issued deployment request identifies the selected web service.

15. (rejected) The method according to claim 11, wherein the issued deployment request provides information about run-time conditions on the selected one of the network locations.

16. (rejected) The method according to claim 8, wherein the replacement comprises executable code.

17. (rejected) The method according to claim 16, wherein the executable code is automatically adapted to the run-time conditions on the selected one of the network locations.

18. (rejected) A system for dynamically redeploying web services in a computing network, comprising:

means for receiving a redeployment trigger for a selected web service wherein the selected web service includes executable code;

means for determining one or more network locations where the selected web service including the executable code has been deployed from its original location at an origin server;

means for programmatically removing the selected web service including the executable code from the network locations and the origin server; and

means for programmatically replacing the selected web service at the network locations and the origin server.

19. (rejected) A computer program product for dynamically redeploying web services in a computing network, the computer program product embodied on one or more computer-readable media and comprising:

computer-readable program code configured to receive a redeployment trigger for a selected web service wherein the selected web service includes executable code;

computer-readable program code configured to determine one or more network locations where the selected web service including the executable code has been deployed from its original location at an origin server;

computer-readable program code configured to programmatically remove the selected web service including the executable code from the network locations and the origin server; and

computer-readable program code configured to programmatically replace the selected web service at the network locations and the origin server.

20. (rejected) The system according to Claim 18 wherein the redeployment trigger comprises a redeployment request from the origin server.

21. (rejected) The computer program product according to Claim 19 wherein the redeployment trigger comprises a redeployment request from the origin server.

22. (rejected) The system according to Claim 18 further comprising:
means for sending the redeployment trigger when the selected web service is to be revised.

23. (rejected) The computer program product according to claim 19 further comprising:
computer-readable program code configured to send the redeployment trigger when the selected web service is to be revised.

24. (rejected) The method according to claim 1:

wherein determining one or more network locations where the selected web service has been deployed includes determining all of the network locations where the selected web service has been deployed;

wherein programmatically removing the selected web service from the network locations includes programmatically removing the selected web service from all of the network location where the web service has been deployed; and

wherein programmatically replacing the selected web service at the network locations includes programmatically replacing the selected web service at all of the network locations where the web service has been deployed.

25. (rejected) The method according to claim 1 wherein programmatically replacing the selected web service at the network locations comprises replacing the selected web service with an updated web service including updated executable code.

26. (rejected) The system according to claim 18:

wherein the means for determining one or more network locations where the selected web service has been deployed includes means for determining all of the network locations where the selected web service has been deployed;

wherein the means for programmatically removing the selected web service from the network locations includes means for programmatically removing the selected web service from all of the network location where the web service has been deployed; and

wherein the means for programmatically replacing the selected web service at the network locations includes programmatically replacing the selected web service at all of the network locations where the web service has been deployed.

27. (rejected) The system according to claim 18 wherein the means for programmatically replacing the selected web service at the network locations comprises means for replacing the selected web service with an updated web service including updated executable code.

28. (rejected) The computer program product according to claim 19:

wherein the computer-readable program code configured to determine one or more network locations where the selected web service has been deployed is further configured to determine all of the network locations where the selected web service has been deployed;

wherein the computer-readable program code configured to programmatically removing the selected web service from the network locations is further configured to programmatically remove the selected web service from all of the network location where the web service has been deployed; and

wherein the computer-readable program code configured to programmatically replacing the selected web service at the network locations is further configured to programmatically replace the selected web service at all of the network locations where the web service has been deployed.

29. (rejected) The computer program product according to claim 19 wherein the computer-readable program code configured to programmatically replace the selected web service at the network locations is further configured to replace the selected web service with an updated web service including updated executable code.

In re: Peter Brittenham et al;
Serial No.: 09/864,608
Filed: May 23, 2001
Page 25 of 26

Appendix B: Evidence

No evidence pursuant to 37 CFR Sec. 1.130, Sec. 1.131, or Sec. 1.132 is relied upon by Appellant in the appeal.

In re: Peter Brittenham et al;
Serial No.: 09/864,608
Filed: May 23, 2001
Page 26 of 26

Appendix C: Related Proceedings

There are no related proceedings pursuant to 37 C.F.R. Sec. 41.37.